MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 2, 2018/2019

DMT5018 – MICROCONTROLLER TECHNOLOGY

(Diploma in Electronic Engineering)

11 MARCH 2019
2:30 PM – 4:30 PM
(2 Hours)

INSTRUCTIONS TO STUDENTS

1. This booklet consists of 8 pages including cover pages with 4 questions only.
2. Attempt **ALL** questions given. All questions carry equal marks and distribution of the marks for each question is given.
3. Please write all your answers in the Answer Booklet provided.
4. All necessary working **MUST** be shown.
5. Opcode map and Special Function Register formats are provided in Appendixes.

QUESTION 1 [25 Marks]

(a) What do these acronyms stand for? Briefly explain the functional description for each of them.

(i) CPU [2 marks]

(ii) ALU [2 marks]

(b) Name TWO of the 8051 control pins. Briefly explain their function. [4 marks]

(c) What is the state of the carry flag, the auxiliary carry flag, the overflow flag, the parity bit and the content of the accumulator after execution of the following instruction sequence? [6 marks]

```
MOV   A, #2
MOV   R6, #8
ADD   A, R6
```

(d) An 8051 microcontroller based system is to be designed requiring 32Kbytes ROM and 32Kbytes RAM, with start-up code stored in ROM. It will also be interfaced to 6 input-output (I/O) devices. Given an available 8Kbytes program memory and 8Kbytes data memory blocks, draw the configuration of this system showing all the 8051 signal lines.

[11 marks]

QUESTION 2 [25 Marks]

(a) List out TWO addressing modes available in an 8051 microcontroller. Give example each. [4 marks]

(b) Answer the following questions:

(i) Briefly describe the operations involved when the instruction "PUSH 30H" is executed. [4 Marks]

(ii) List out bit addresses that are set to 1 as a result of the following instructions?

MOV 29H, #25H [2 marks]

(iii) Write an 8051 assembly language program to add the contents from memory location 44H and R6 respectively. The result should be stored in memory location 44H. [3 Marks]

(iv) Find the respective assembly language instruction for the following machine code: F5H, 66H. [2 marks]

Continued...

(c) Assembly language instructions for the 8051 microcontroller is given below:

```
        MOV     R2,#01H
LOOP:   MOV     R1, #80H
LOOP1:  MOV     @R1, #0FH
        DEC     R1
        CJNE    R1, #20H, LOOP1
        MOV     R0, #90H
LOOP2:  MOV     @R0, #0F0H
        INC     R0
        CJNE    R0, @0F0H, LOOP2
        DEC     R2
        CJNE    R2, #00H, LOOP
        RET
```

Find the following:

(i) The total memory requirement in bytes. [2 Marks]

(ii) The affected memory locations (after execution) with their final values. [4 Marks]

(iii) Total time to execute if 12MHz clock is used for the 8051 microcontroller. [4 Marks]

QUESTION 3 [25 Marks]

(a) Name the special function registers associated with Timer operation of the 8051 microcontroller. [4 Marks]

(b) A system is designed (using Timer 1 of 8051 microcontroller) to count and to indicate the capacity of a hall. A sensor generates a pulse to T1 (Timer 1 input) each time a person enters the hall. If the hall is full (capacity of 800), a light connected to P1.0 should be on. Write an assembly language instruction sequence for 8051 microcontroller to implement the system. [8 Marks]

(c) Write an assembly language programme to initialize serial port for 8-bit UART communication with a baud rate of 2400 when using an 11.059MHz crystal and Timer 1. (Hint: Use SMOD = 0). [5 Marks]

(d) Write an assembly language programme to generate a pulse width modulation signal on P1.2 with 60ms period and 10% duty cycle using Timer 1 with 12MHz crystal. [8 Marks]

Continued .....

QUESTION 4 [25 Marks]

(a) Answer the following questions:

(i) Define interrupt service routine (ISR) [2 marks]

(ii) What is the difference between small ISRs and large ISRs? [2 marks]

(b) Following instructions are executed by an 8051 microcontroller:

```
MOV  IE, #87H
MOV  IP, #06H
```

(i) Which interrupts are enabled? [3 marks]

(ii) Which interrupt has the highest priority? [1 mark]

(iii) What is the sequence of servicing the interrupts if all enabled interrupts occur simultaneously? [3 marks]

(c) Four **common anode** seven-segment display are connected to the 8051 microcontroller at Port 1 via a latch driver, 74LS07 while P2.0 to P2.3 are used to drive the common anodes.

(i) Sketch the diagram showing the connection of the 8051 and the seven-segment display. [4 marks]

(ii) Write the values in 8-bit binary and hexadecimal format for the character display of 'A', 'C', 'E' and '0'. [4 marks]

(iii) Write a program which will create the seven-segment display A → C → E → 0, in continuous manner, with delay between each character transition determined by the following DELAY_5 subroutine. [6 marks]

```
DELAY_5:  MOV  R1, #10
          MOV  R0, #0
LOOP:     DJNZ R0, LOOP
          DJNZ R1, LOOP
          RET
```

Continued .....

Appendix A: Opcode Map

byte / Instruction operands / cycle	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1B NOP 1C	3B JBC bit, rel 2C	3B JB bit, rel 2C	3B JNB bit, rel 2C	2B JC rel 2C	2B JNC rel 2C	2B JZ rel 2C	2B JNZ rel 2C	2B SJMP rel 2C	3B MOV DPTR, #data16 2C	2B ORL C, /bit 2C	2B ANL C, /bit 2C	2B PUSH dir 2C	2B POP dir 2C	1B MOVX A, @DPTR 2C	1B MOVX @DPTR, A 2C
1	2B AJMP (P0) 2C	2B ACALL (P0) 2C	2B AJMP (P1) 2C	2B ACALL (P1) 2C	2B AJMP (P2) 2C	2B ACALL (P2) 2C	2B AJMP (P3) 2C	2B ACALL (P3) 2C	2B AJMP (P4) 2C	2B ACALL (P4) 2C	2B AJMP (P5) 2C	2B ACALL (P5) 2C	2B AJMP (P6) 2C	2B ACALL (P6) 2C	2B AJMP (P7) 2C	2B ACALL (P7) 2C
2	3B LJMP addr16 2C	3B LCALL addr16 2C	1B RET 2C	1B RETI 2C	2B ORL dir, A 1C	2B ANL dir, A 1C	2B XRL dir, A 1C	2B ORL C, bit 2C	2B ANL C, bit 2C	2B MOV bit, C 2C	2B MOV C, bit 1C	2B CPL bit 1C	2B CLR bit 1C	2B SETB bit 1C	1B MOVX A, @R0 2C	1B MOVX @R0, A 2C
3	1B RR A 1C	1B RRC A 1C	1B RL A 1C	1B RLC A 1C	3B ORL dir, #data 2C	3B ANL dir, #data 2C	3B XRL dir, #data 2C	1B JMP @A+DPTR 2C	1B MOVC A, @A+PC 2C	1B MOVC A, @A+DPTR 2C	1B INC DPTR 2C	1B CPL C 1C	1B CLR C 1C	1B SETB C 1C	1B MOVX A, @R1 2C	1B MOVX @R1, A 2C
4	1B INC A 1C	2B DEC A 1C	2B ADD A, #data 1C	2B ADDC A, #data 1C	2B ORL A, #data 1C	2B ANL A, #data 1C	2B XRL A, #data 1C	2B MOV A, #data 1C	1B DIV AB 4C	2B SUBB A, #data 1C	1B MUL AB 4C	3B CJNE A, #data, rel 2C	1B SWAP A 1C	1B DA A 1C	1B CLR A 1C	1B CPL A 1C
5	2B INC dir 1C	2B DEC dir 1C	2B ADD A, dir 1C	2B ADDC A, dir 1C	2B ORL A, dir 1C	2B ANL A, dir 1C	2B XRL A, dir 1C	3B MOV dir, #data 2C	2B MOV dir, dir 2C	2B SUBB A, dir 1C	[redacted]	3B CJNE A, dir, rel 2C	2B XCH A, dir 1C	3B DJNZ dir, rel 2C	2B MOV A, dir 1C	2B MOV dir, A 1C
6	1B INC @R0 1C	1B DEC @R0 1C	1B ADD A, @R0 1C	1B ADDC A, @R0 1C	1B ORL A, @R0 1C	1B ANL A, @R0 1C	1B XRL A, @R0 1C	2B MOV @R0, #data 1C	2B MOV dir, @R0 2C	1B SUBB A, @R0 1C	2B MOV @R0, dir 2C	3B CJNE @R0, #data,rel 2C	1B XCH A, @R0 1C	1B XCHD A, @R0 1C	1B MOV A, @R0 1C	1B MOV @R0, A 1C
7	1B INC @R1 1C	1B DEC @R1 1C	1B ADD A, @R1 1C	1B ADDC A, @R1 1C	1B ORL A, @R1 1C	1B ANL A, @R1 1C	1B XRL A, @R1 1C	2B MOV @R1, #data 1C	2B MOV dir, @R1 2C	1B SUBB A, @R1 1C	2B MOV @R1, dir 2C	3B CJNE @R1, #data,rel 2C	1B XCH A, @R1 1C	1B XCHD A, @R1 1C	1B MOV A, @R1 1C	1B MOV @R1, A 1C
8	1B INC R0 1C	1B DEC R0 1C	1B ADD A, R0 1C	1B ADDC A, R0 1C	1B ORL A, R0 1C	1B ANL A, R0 1C	1B XRL A, R0 1C	2B MOV R0, #data 1C	2B MOV dir, R0 2C	1B SUBB A, R0 1C	2B MOV R0, dir 2C	3B CJNE R0, #data, rel 2C	1B XCH A, R0 1C	2B DJNZ R0, rel 2C	1B MOV A, R0 1C	1B MOV R0, A 1C
9	1B INC R1 1C	1B DEC R1 1C	1B ADD A, R1 1C	1B ADDC A, R1 1C	1B ORL A, R1 1C	1B ANL A, R1 1C	1B XRL A, R1 1C	2B MOV R1, #data 1C	2B MOV dir, R1 2C	1B SUBB A, R1 1C	2B MOV R1, dir 2C	3B CJNE R1, #data, rel 2C	1B XCH A, R1 1C	2B DJNZ R1, rel 2C	1B MOV A, R1 1C	1B MOV R1, A 1C
A	1B INC R2 1C	1B DEC R2 1C	1B ADD A, R2 1C	1B ADDC A, R2 1C	1B ORL A, R2 1C	1B ANL A, R2 1C	1B XRL A, R2 1C	2B MOV R2, #data 1C	2B MOV dir, R2 2C	1B SUBB A, R2 1C	2B MOV R2, dir 2C	3B CJNE R2, #data, rel 2C	1B XCH A, R2 1C	2B DJNZ R2, rel 2C	1B MOV A, R2 1C	1B MOV R2, A 1C
B	1B INC R3 1C	1B DEC R3 1C	1B ADD A, R3 1C	1B ADDC A, R3 1C	1B ORL A, R3 1C	1B ANL A, R3 1C	1B XRL A, R3 1C	2B MOV R3, #data 1C	2B MOV dir, R3 2C	1B SUBB A, R3 1C	2B MOV R3, dir 2C	3B CJNE R3, #data, rel 2C	1B XCH A, R3 1C	2B DJNZ R3, rel 2C	1B MOV A, R3 1C	1B MOV R3, A 1C
C	1B INC R4 1C	1B DEC R4 1C	1B ADD A, R4 1C	1B ADDC A, R4 1C	1B ORL A, R4 1C	1B ANL A, R4 1C	1B XRL A, R4 1C	2B MOV R4, #data 1C	2B MOV dir, R4 2C	1B SUBB A, R4 1C	2B MOV R4, dir 2C	3B CJNE R4, #data, rel 2C	1B XCH A, R4 1C	2B DJNZ R4, rel 2C	1B MOV A, R4 1C	1B MOV R4, A 1C
D	1B INC R5 1C	1B DEC R5 1C	1B ADD A, R5 1C	1B ADDC A, R5 1C	1B ORL A, R5 1C	1B ANL A, R5 1C	1B XRL A, R5 1C	2B MOV R5, #data 1C	2B MOV dir, R5 2C	1B SUBB A, R5 1C	2B MOV R5, dir 2C	3B CJNE R5, #data, rel 2C	1B XCH A, R5 1C	2B DJNZ R5, rel 2C	1B MOV A, R5 1C	1B MOV R5, A 1C
E	1B INC R6 1C	1B DEC R6 1C	1B ADD A, R6 1C	1B ADDC A, R6 1C	1B ORL A, R6 1C	1B ANL A, R6 1C	1B XRL A, R6 1C	2B MOV R6, #data 1C	2B MOV dir, R6 2C	1B SUBB A, R6 1C	2B MOV R6, dir 2C	3B CJNE R6, #data, rel 2C	1B XCH A, R6 1C	2B DJNZ R6, rel 2C	1B MOV A, R6 1C	1B MOV R6, A 1C
F	1B INC R7 1C	1B DEC R7 1C	1B ADD A, R7 1C	1B ADDC A, R7 1C	1B ORL A, R7 1C	1B ANL A, R7 1C	1B XRL A, R7 1C	2B MOV R7, #data 1C	2B MOV dir, R7 2C	1B SUBB A, R7 1C	2B MOV R7, dir 2C	3B CJNE R7, #data, rel 2C	1B XCH A, R7 1C	2B DJNZ R7, rel 2C	1B MOV A, R7 1C	1B MOV R7, A 1C

Continued .....

Appendix B: Special Function Register (SFR)

Byte address	Bit address								
FF									
F0	F7	F6	F5	F4	F3	F2	F1	F0	B
E0	E7	E6	E5	E4	E3	E2	E1	E0	ACC
D0	D7	D6	D5	D4	D3	D2	D1	D0	PSW
B8	-	-	-	BC	BB	BA	B9	B8	IP
B0	B7	B6	B5	B4	B3	B2	B1	B0	P3
A8	AF	-	-	AC	AB	AA	A9	A8	IE
A0	A7	A6	A5	A4	A3	A2	A1	A0	P2
99	not bit addressable								SBUF
98	9F	9E	9D	9C	9B	9A	99	98	SCON
90	97	96	95	94	93	92	91	90	P1
8D	not bit addressable								TH1
8C	not bit addressable								TH0
8B	not bit addressable								TL1
8A	not bit addressable								TL0
89	not bit addressable								TMOD
88	8F	8E	8D	8C	8B	8A	89	88	TCON
87	not bit addressable								PCON
83	not bit addressable								DPH
82	not bit addressable								DPL
81	not bit addressable								SP
80	87	86	85	84	83	82	81	80	P0

Continued .....

Appendix C: Bit-Addressable Ram Locations

Byte Address	Bit Address							
2F	7F	7E	7D	7C	7B	7A	79	78
2E	77	76	75	74	73	72	71	70
2D	6F	6E	6D	6C	6B	6A	69	68
2C	67	66	65	64	63	62	61	60
2B	5F	5E	5D	5C	5B	5A	59	58
2A	57	56	55	54	53	52	51	50
29	4F	4E	4D	4C	4B	4A	49	48
28	47	46	45	44	43	42	41	40
27	3F	3E	3D	3C	3B	3A	39	38
26	37	36	35	34	33	32	31	30
25	2F	2E	2D	2C	2B	2A	29	28
24	27	26	25	24	23	22	21	20
23	1F	1E	1D	1C	1B	1A	19	18
22	17	16	15	14	13	12	11	10
21	0F	0E	0D	0C	0B	0A	09	08
20	07	06	05	04	03	02	01	00

Appendix D: Program Status Word (PSW)

CY	AC	F0	RS1	RS0	OV	--	P

Appendix E: TCON (Timer/Counter Control Register)

TF1	TR1	TF0	TR1	IE1	IT1	IE0	IT0
7	6	5	4	3	2	1	0

Appendix F: TMOD

GATE	C/T	M1	M0	GATE	C/T	M1	M0
7	6	5	4	3	2	1	0

Appendix G: IE (Interrupt Enable Register)

EA	--	ET2	ES	ET1	EX1	ET0	EX0
7	6	5	4	3	2	1	0

Appendix H: IP (Interrupt Priority Register)

--	--	PT2	PS	PT1	PX1	PT0	PX0
7	6	5	4	3	2	1	0

Continued .....

Appendix I: Interrupt Vectors

Interrupt	Flag	Vector Address
Reset	RST	0000H
External 0	IE0	0003H
Timer 0	TF0	000BH
External 1	IE1	0013H
Timer 1	TF1	001BH
Serial Port	RI or TI	0023H

Appendix J: SCON (Serial Port Control Register)

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
7	6	5	4	3	2	1	0

End of Paper